

Types de données

Tristan Crolard

Laboratoire CEDRIC
Equipe « Systèmes Sûrs »

`tristan.crolard@cnam.fr`

`cedric.cnam.fr/sys/crolard`

Records – dataclasses

- ▶ **Dataclasses** are built upon object-oriented features: they are defined as a special kind of classes, using the decorator `@dataclass`
- ▶ However, they behave like **named tuples** (not like objects): they are similar to **records** and **structs** from other programming languages.
- ▶ **Pattern matching** is also available for dataclasses.
- ▶ They can be used to define **variants** or **recursive datatypes**

Dataclass basic example – Rational

```
from dataclasses import dataclass

@dataclass
class Rational:
    num: int
    denom: int

def display(r: Rational) -> None:
    match r:
        case Rational(num=x, denom=y):
            print("num_□=", x, ";", "denom_□=", y)

r = Rational(3, 7)
print(r)
print(r.num)
print(r.denom)
display(r)
```

Variant type example – Value

```
from dataclasses import dataclass

type primitive = IntValue | BoolValue | StrValue

@dataclass
class IntValue:
    value: int

@dataclass
class BoolValue:
    value: bool

@dataclass
class StrValue:
    value: str
```

Dataclass variant example – pattern matching

```
def to_str(v: primitive) -> str:  
  match v:  
    case IntValue(value=i):  
      return str(i)  
    case BoolValue(value=b):  
      return str(b)  
    case StrValue(value=s):  
      return s
```

Dataclass recursive example – Tree

```
from dataclasses import dataclass
```

```
type tree[A] = Empty[A] | Node[A]
```

```
@dataclass
```

```
class Empty[A]:
```

```
    pass
```

```
@dataclass
```

```
class Node[A]:
```

```
    left: tree[A]
```

```
    value: A
```

```
    right: tree[A]
```

Dataclass recursive example – pattern matching

```
def size[A](t: tree[A]) -> int:  
  match (t):  
    case Empty():  
      return 0  
    case Node(left=l, value=_, right=r):  
      return size(l) + 1 + size(r)
```

```
t: tree[int] = Node(Empty(), 5, Node(Empty(), 3, Empty()))  
print(t)  
print(size(t))
```